

**Программный проект. НИУ ВШЭ, 2 курс**

Прикладная математика и информатика.

Сервис по распознаванию эмоций человека.  
Королев Кирилл Андреевич.

Научный руководитель: Доцент, факультет компьютерных наук ВШЭ.  
Кантонистова Елена Олеговна.

Москва 2022

# 1 Цели и мотивация проекта

Изучить различные модели машинного обучения и архитектуры нейронных сетей для решения задачи по распознаванию эмоций человека по изображению.

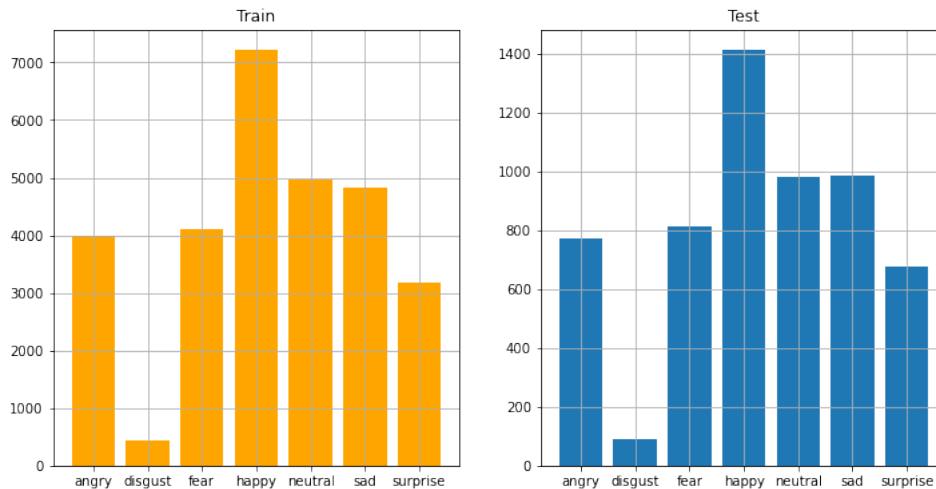
Для реализации поставленной цели требуется выполнить ряд задач: провести ресерч по моделям машинного обучения и подходам в решении поставленной задачи, обработать данные, взятые из открытых источников, и обучить модель на них и реализовать веб-сервис с заявленной функциональностью.

## 2 Данные

Датасет для обучения взят с сайта Kaggle. Он представляет из себя 2 директории train и test для обучения и тестирования соответственно. Каждая из них содержит директории под эмоции: happy, neutral, sad, fear, angry, surprise, disgust, с изображениями размера 48x48.



Распределение классов для обучающей и тестовой выборки выглядит следующим образом.



Видим, что распределения примерно одинаковые и есть надежда рассчитывать на удачные результаты. Неравномерный баланс классов учтем при дальнейшем обучении.

Также будем использовать так называемую валидационную выборку, на которой будем перебирать гиперпараметры для нашей модели.

## 3 Модели

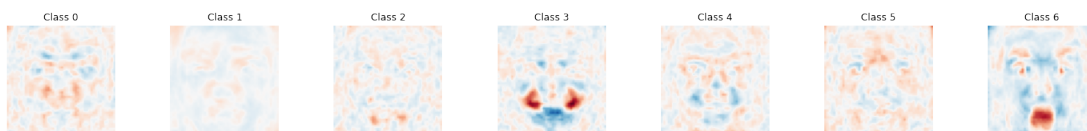
### 3.1 Мультиклассовая классификация

Допустим у нас есть бинарный классификатор. Возникает вопрос, как построить много-классовый классификатор на его основе. Существует несколько подходов, например, One vs Rest, когда мы обучаем  $L$  бинарных классификаторов для каждого класса и выбираем тот класс, на котором вероятность максимальна. Он в итоге и показал наилучшие результаты.

### 3.2 Логистическая регрессия

Мы хотим предсказывать вероятности классов для изображения. Для мультиклассовой классификации выбор падает на логистическую регрессию, либо на метод опорных векторов. Фичами (или признаками) будут сами пиксели изображения.

Используем логистическую регрессию с L2-регуляризацией, то есть в функцию потерь добавляем штраф за большую вторую норму вектора параметров модели. Замечательное свойство логистической регрессии, что ее параметры интерпретируемы. Построим графики для каждого класса.



Видим характерные черты лица для некоторых классов, например, для disgust, happy и surprise довольно четко видны очертания, которые свойственны этим эмоциям.

### 3.3 SVM

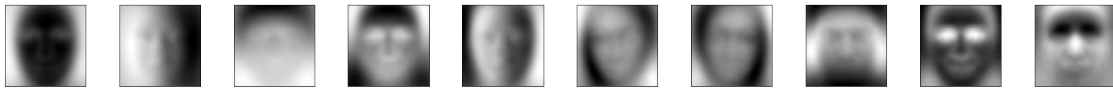
В методе опорных векторов существует подход, что если классы не обладают линейной разделимостью, то можно использовать так называемые ядра для создания нелинейного классификатора. Лучше всего себя показала rbf - радиальная базисная функция.

Постараемся улучшить качество.

### 3.4 PCA

Попробуем уменьшить размерность, чтобы увеличить скорость обучения, разложив наши изображения по некоторому базису главных компонент. PCA по сути сводится к сингулярному разложению матрицы данных. Опытным путем установили, что лучшая точность достигается при ранге равном 250.

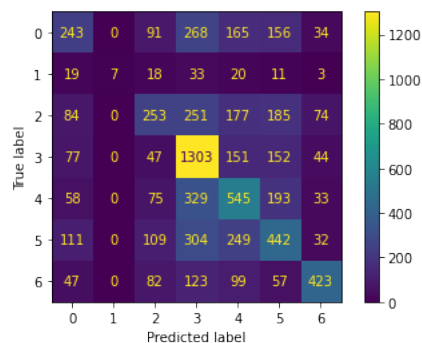
Несколько первых компонент.



Точность (ассигасу) при этом значительно улучшилась.

### 3.5 Аугментация классов

Посмотрим на матрицу ошибок. На главной диагонали расположено количество правильных предсказаний для каждого класса.



Выберем классы, на которых больше всего ошибок. Возьмем изображения из этих классов и, например, отразим их и добавим в выборку, тогда количество сэмплов увеличится вдвое. Наблюдения показывают, что тогда ошибки на этих классах должны уменьшиться. Можно делать и другие преобразования с изображениями.

### 3.6 CNN

Несмотря на улучшения, ассигасу не превосходила 50%. Возникла идея использовать сверточные нейронные сети, которые лучше всего показывают себя при работе с изображениями.

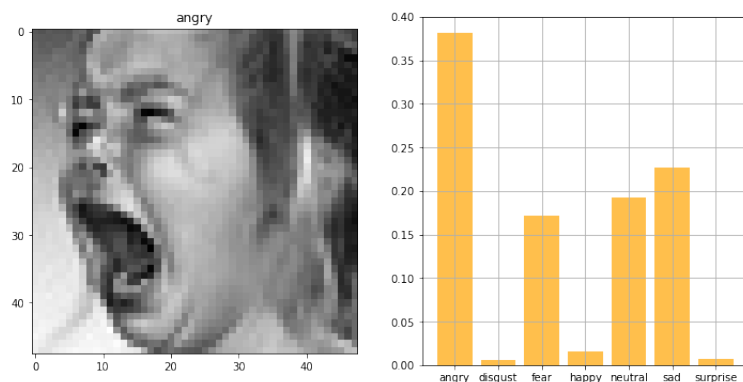
Архитектура нейронной сети представляет из себя последовательные комбинации следующих слоев: операции свертки, активации, пулинга (pooling) и дропаута (dropout).

Слой свёртки включает в себя свой фильтр, ядро свёртки которого обрабатывает предыдущий слой по фрагментам. Результат каждой свёртки попадает на функцию активации, которая представляет собой нелинейную функцию. Чаще всего используют функцию  $ReLU$   $f(x) = \max(0, x)$ . За счет своей простоты она значительно ускоряет скорость обучения.

Если на предыдущей операции свёртки уже были выявлены некоторые признаки, то для дальнейшей обработки настолько подробное изображение уже не нужно, и оно уплотняется до менее подробного. Для этого и предназначен пулинг.

Dropout в свою очередь предназначен для уменьшения переобучаемости нейронной сети.

На выходе пользуемся софтмаксом, чтобы получить вероятности для каждого класса. Качество выросло с 50% при использовании SVM до почти 60% с CNN.



Пример классификации изображения из тестовой выборки.

## 4 Метрики

Особо важно определиться с метриками, чтобы делать выводы о работоспособности моделей и алгоритмов. Пока что до этого речь шла только об ассигасу, однако она довольно бесполезна, когда мы имеем дело с несбалансированными классами. Поэтому при классификации чаще используют precision и recall (или их комбинации).

Для мультиклассовой классификации precision считается как сумма всех true positive объектов среди всех классов, деленная на сумму true positive и false positive тоже среди всех классов. Recall в свою очередь подсчитывает долю правильных положительных предсказаний среди всех положительных предсказаний.

Итоговые результаты приведены ниже.

	raw log regression	SVM with rbf kernel + PCA	CNN
accuracy	0.36	0.45	57.55
precision	0.33	0.45	63.93
recall	0.36	0.45	50.75

## 5 Веб-сервис


### Emocl


Choose your image for classification


Выберите файл


PrivateTest\_2264189.jpg


Classify


  
**Angry**  
0.33%


  
**Disgust**  
0.00%

  
**Fear**  
2.84%

  
**Happy**  
0.37%

  
**Neutral**  
0.01%

  
**Sad**  
0.01%

  
**Surprise**  
96.45%

Сервис написан на Python в связке с Django для бэкенда и JS для фронтенда. Пользователь может выбрать изображение на своем компьютере, и по нажатию кнопки посмотреть на вероятности эмоций человека на фотографии. Уже обученная модель нейронной сети хранится на сервере в бинарном виде.

## 6 Ссылки

Демо можно найти по ссылкам в репозитории.

- Репозиторий с кодом
- Ноутбук 1
- Ноутбук 2